Remarks

In the Office action dated March 22, 2006("Office action"), all presently pending claims (claims 18-50) were rejected under both the judicially created doctrine of double patenting in view of U.S. Patent No. 6,353,923, and as anticipated by U.S. Patent No. 5,940,593 to House et al. ("House") pursuant to § 102(e). Applicants respectfully traverse these rejections.

1. The Double Patenting Rejection of Claims 18-50 Is Moot.

In response to the double patenting rejection, Applicants by their attorney file the attached terminal disclaimer, as both the present application and the '923 patent are owned by the same assignee. Applicants wish to emphasize that the filing of this terminal disclaimer is solely for purposes of moving this application forward to issuance. Applicants in no way concede or agree with the Examiner's rejection.

2. House Does Not Teach or Suggest at Least One Limitation of Each of Claims 18-50.

House describes a system for emulating a three-tier computer system on a single machine for purposes of debugging applications that would ordinarily be run over a network. The first tier includes a computer that runs a web browser. (House, 3:50-:56.) The third tier includes a database server and VAB-II runtime module that executes stored procedure ("SP") and user defined function ("UDF") scripts and receives commands from tier 2 for accessing the database. (*Id.*, 4:7-:20.) The second tier includes a web server and a VAB-II runtime module for executing scripts. House describes the scripts executed by the VAB-II runtime module(s) as coming from local APP file(s). (*Id.*, 3:57-4:6; 5:55-:60.)

According to House, scripts in the second tier can contain programming logic for communicating with a browser and the third tier. (*Id.*, 3:67-4:3.) House repeatedly describes the second-tier scripts and third-tier scripts as being LotusScript scripts (*id.*, 4:1, 4:10-:13, 5:54-6:6, 6:32-49) and briefly describes script interactivity with external controls and databases (*id.*, 4:3-6). As detailed in the LotusScript documentation submitted herewith, LotusScript is a version of BASIC, and LotusScript scripts include BASIC-like instructions. (Programmer's Guide, LotusScript Release 3, page 1-1.) LotusScript supports interaction with objects outside a script through Dynamic Data Exchange, OLE automation, and other technologies.

(Programmer's Guide, Chapter 9.) LotusScript also supports use of embedded SQL statements. (Approach User Support, "Using SQL from LotusScript.") However, LotusScript itself is a single language (a version of BASIC, as mentioned above).

Likewise, House describes debugging with LotusScript tools: "One or more LotusScript IDEs are executed, wherein each IDE includes a code window with a debugger." (*Id.*, 5:61-:62.) "[A]s earlier described, first second and third tiers are defined on the development computer **400**. . . . [B]oth the second and third tiers comprise the VAB-II runtime engine **116, 128**. This provides a common codebase, and allows application debugging of the LotusScript logic to be the same for both tiers." (*Id.*, 5:30-:31, 5:33-:37.) For example, a user interacts with controls, which results in interaction with the scripts. "As the user interacts with controls in the HTML form displayed by the browser 108, the application script 118 and SP/UDF script 130 are executed and breakpoints with the scripts are triggered."

House fails to teach or suggest at least one limitation in each of claims 18-50.

Claim 18: ... providing a debugging environment for debugging mixed-language script, the mixed-language script interacting with features of a host through a programming interface exposed by the host, the mixed-language script including a first script portion written in a first language and a second script portion written in a second language;
recognizing a debuggable entity created from the mixed-language script and context information;

Claim 26: ... a debuggable entity created from mixed-language script and context information, the mixed-language script for interacting with features of a host through a programming interface exposed by the host, the mixed-language script including a first script portion written in a first language and a second script portion written in a second language;

Claim 27: ... providing a debugging environment for debugging mixed-language script that interacts with features of a web browser and with features of a remote host, the mixed-language script including a first script portion written in a first language and a second script portion written in a second language;
recognizing a debuggable entity created from the mixed-language script and context information;

Claim 31: ... a language-independent host for hosting mixed-language script that interacts with features of the host, the mixed-language script including a first script portion written in a first language and a second script portion written in a second language;

Claim 37: ... receiving a language-independent description of a debugging activity related to mixed-language script that interacts with features of a host, the mixed-language script including a first script portion written in a first language and a second script portion written in a second language;

Claim 48: ... requesting a first language engine to enumerate first contents of a first stack frame, the first language engine supporting language-dependent implementation according to a first language, the first contents including first language-dependent stack frame information;

requesting a second language engine to enumerate second contents of a second stack frame, the second language engine supporting language-dependent implementation according to a second language, the second contents including second language-dependent stack frame information; and

aggregating the first contents and the second contents.

House does not teach or suggest the above-cited language of claims 18, 26, 27, 31, 37 and 48, respectively. Each of claims 18, 26, 27, 31, and 37 includes the limitation, "the mixed-language script including a first script portion written in a first language and a second script portion written in a second language." Likewise, claim 48 requires two distinct language engines enumerating stack frames, each engine supporting a language-dependent implementation according to a different language. In contrast, House describes executing and debugging LotusScript scripts (namely, the application script 118 and SP/UDF script 130) using one or more LotusScript IDEs. This involves executing/debugging scripts having BASIC-like instructions that may interact with controls and databases. Executing/debugging LotusScript scripts (as in House), which are single script language scripts, leads away from the above-cited language of claims 18, 26, 27, 31, 37 and 48, respectively.

The Examiner writes, "HTML is one of the languages used, which inherently has mixed language scripts (html, javascript, etc.)." (Office action, page 6.) The Applicants disagree with the Examiner's mapping of this language from House to the "mixed-language script" language of claims 18, 26, 27, 31 and 37, respectively. While HTML can be used to specify pages with forms and has various other features, use of HTML (as in House) does not imply the presence of mixed-language scripts.

Additionally, claims 18 and 27 require the additional step of recognizing a debuggable entity created from a mixed-language script and context information. Claim 26 recites, "a

debuggable entity created from mixed-language script and context information." As noted above, House does not teach or suggest the "mixed-language script" language of claims 18, 26, and 27, respectively. House is even further from teaching or suggesting a debuggable entity created from mixed-language script and context information as recited in claims 18, 26 and 27, respectively.

Claim 37 recites, "coordinating implementation of the debugging activity through a language engine that handles language-dependent execution and debugging for the debugging activity." As noted above, House does not teach or suggest the "mixed-language script" language of claim 37. House is even further from teaching or suggesting a language engine that handles language-dependent execution and debugging for the debugging activity as recited in claim 37.

Claim 48 recites, "first contents including first language-dependent stack frame information, "second contents including second language-dependent stack frame information" and "aggregating the first contents and the second contents." As noted above, House does not teach or suggest the "language-dependent implementation" language of claim 48. House is even further from teaching or suggesting aggregating first contents and second contents as recited in claim 48.

For at least these reasons, claims 18, 26, 27, 31, 37 and 48 should be allowable.

Dependent claims 19-25 depend from independent claim 18, dependent claims 28-30 depend from independent claim 27, dependent claims 32-36 depend from independent claim 31, dependent claims 38-47 depend from claim 37, and dependent claims 49 and 50 depend from independent claim 48. As all independent claims are not anticipated by House and are in condition for allowance, the separate patentability of the dependent claims will not be belabored here.

As all claims are now in condition for allowance, such action is respectfully requested.

Respectfully submitted,

KLARQUIST SPARKMAN, LLP

By _____
Jonathan M. Ward
Registration No. 56,466

One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Telephone: (503) 595-5300
Facsimile: (503) 228-9446